A SystemC-AMS Framework for the Design and Simulation of Energy Management in Electric Vehicles

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 28, 2019, accepted February 11, 2019, date of publication February 21, 2019, date of current version March 8, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2900505*

# A SystemC-AMS Framework for the Design and Simulation of Energy Management in Electric Vehicles

**YUKAI CHEN[1], (Member, IEEE), DONKYU BAEK[1], (Member, IEEE), JAEMIN KIM[2], (Member, IEEE), SANTA DI CATALDO[1], (Member, IEEE), NAEHYUCK CHANG[3], (Fellow, IEEE), ENRICO MACII[4], (Fellow, IEEE), SARA VINCO[1], (Member, IEEE), AND MASSIMO PONCINO[1], (Fellow, IEEE)**

[1]Department of Control and Computer Engineering, Politecnico di Torino, 10129 Turin, Italy
[2]Department of Electronics Engineering, Myongji University, Yongin 03674, South Korea
[3]School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea
[4]Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, 10129 Turin, Italy

Corresponding author: Yukai Chen (yukai.chen@polito.it)

**ABSTRACT** Driving range is one of the most critical issues for electric vehicles (EVs): running out of battery charge while driving results in serious inconvenience even comparable to a vehicle breakdown, as an effect of long fuel recharging times and lack of charging facilities. This may discourage EVs for current and potential customers. As an effect, the dimensioning of the energy subsystem of an EV is a crucial issue: the choice of the energy storage components and the policies for their management should be validated at design time through simulations, so to estimate the vehicle driving range under reference driving profiles. Thus, it is necessary to build a simulation framework that considers an EV power consumption model that accounts for the characteristics of the vehicle and the driving route, plus accurate models for all power components, including batteries and renewable power sources. The goal of this paper is to achieve such an early EV simulation, through the definition of a SystemC-AMS framework, which models simultaneously the physical and mechanical evolution, together with energy flows and environmental characteristics. The proposed solution extends the state-of-the-art framework for the simulation of electrical energy systems with support for mechanical descriptions and the AC domain, by finding a good balance between accuracy and simulation speed and by formalizing the new information and energy flows. The experimental results demonstrate that the performance of the proposed approach in terms of accuracy and simulation speed w.r.t. the current state-of-the-art and its effectiveness at supporting EV design with an enhanced exploration of the alternatives.

**INDEX TERMS** Cyber-physical systems, design-time optimization, electric vehicles, electrical energy system, SystemC-AMS.

## I. INTRODUCTION

Electric Vehicles (EVs) are posing new challenges for Cyber Physical System (CPS) design: the introduction of components such as batteries and energy management policies imply indeed a tighter interaction between mechanical components, computational units and the physical environment, including the grid [1], [2].

As a matter of fact, each EV comprises a number of interacting CPSs. Compared to the combustion engine vehicles, the most relevant CPS of an EV is the powertrain, i.e., the EV energy subsystem, that includes battery, electric motor, energy management systems and the necessary inverters and converters. The design of the energy subsystem is thus a critical design dimension for EVs. The main limitations to their widespread adoption are: (1) to their limited autonomy, due to low efficiency, (2) to the still limited presence of charging facilities, and (3) to the long charging times [3].

As a result, during the design flow of an EV it is necessary to carefully choose and dimension the energy sources, to ensure long driving cycles and to reduce conversion inefficiencies. In this perspective, computer modeling and

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Anvari-Moghaddam.

simulation are a key resource for designers: testing the behavior of the vehicle on reference routes allows indeed to estimate battery lifetime, and to track the behavior of all energy components, so to take into account non-ideal behaviors and conversion losses [4]. Additionally, the simulation framework should allow easy reconfiguration, to explore different component configurations and energy management strategies, before the EV prototype is ready.

The high interest in the design of EVs generated a number of simulation approaches in the literature. However, most works focus on power consumption estimation, with limited or no modeling of the energy subsystem [5]. Additionally, the models for power consumption accurately reproduce the mechanics of the EV, thus resulting in long and complex simulations of the internal components of the motor [6]–[8]. This makes day-long simulations unfeasible, while they are on the other hand necessary to estimate the EV driving range and to validate the dimensioning of the energy components.

This work faces this challenge by simulating the energy subsystem of an EV through the definition of a SystemC-AMS framework. The modularity of SystemC-AMS and its support for multiple models of computation allow to easily cover all domains included in an EV with a lightweight yet accurate simulation. The framework extends the solution proposed in [9] for generic electrical energy systems.

The main novel contributions of this work are:

- The formalization of the information and energy flows, and of the components typically involved in the simulation of the energy subsystem of an EV;
- The modeling of the alternating current (AC) domain in SystemC-AMS, by reaching a compromise between the complex sinusoidal nature of AC and the need for simulation speed of the framework;
- The identification of a suitable model of power consumption, to accurately simulate EV power consumption preserving to some extent the mechanical aspects, still without reproducing in detail the behavior of motor components or killing simulation speed;
- The application to a Tesla Model 3 EV, to compare the proposed framework w.r.t. the current state-of-the-art in terms of simulation speed and accuracy;
- The development of a photovoltaic (PV) EV, i.e., a small EV equipped with solar panels, used for design space exploration of alternative configurations.

The paper is organized as follows. Section II provides the necessary background. Section III details the proposed SystemC-AMS framework. Section IV shows how to generate the input driving traces. Finally, Section V focuses on the experimental results, and Section VI draws our conclusions.

## II. BACKGROUND

This Section introduces the main solutions for EV simulation at state-of-the-art (Section II-A) and the main characteristics of the SystemC-AMS standard for analog and mixed signal modeling (Section II-B).

### A. SIMULATION OF EVs

EV modeling has been widely investigated in the literature. *ADVISOR* is the main reference for the design electric vehicles or hybrid electric vehicles in terms of vehicle performance, fuel economy and emissions [10], [11]. ADVISOR provides a library of models reproducing the dynamics and the mechanical evolution of each internal component, that can be used to configure the vehicle and the energy subsystem. From this configuration, ADVISOR estimates energy consumption under a given driving profile by determining vehicle dynamics, the corresponding required motor power and the related battery current.

Other MATLAB-based approaches use a similar library-based strategy, with differences in terms of simulation speed or of how they determine vehicle dynamics. PSAT (Powertrain System Analysis Toolkit) estimates mechanical evolution (e.g., vehicle speed and motor torque) from current load and vehicle dynamics. This increases accuracy, but it leads to slow simulation times [12]. SIMPLEV simulates fuel economy and power consumption under input driving profiles, and it models each component in terms of corresponding power loss in the drivetrain [13]. ELPH studies peak power and control schemes of hybrid EVs [4], [14], and it supports simple models of battery current and voltage.

*Modelica* allows to design vehicle specification to estimate EV performance and energy consumption with an equation-based approach, including in the analysis external disturbances like side wind force and longitudinal acceleration [15]. This is however out of our research scope.

All the aforementioned Matlab/Simulink and Modelica-based solutions target the mechanics of the EV, thus resulting in long and complex simulations that require very detailed models of the internal components of the motor, represented as complex differential equations [6]–[8]. This makes day-long simulations unfeasible, thus limiting design space exploration and reducing the effectiveness of validation of the configuration chosen for the energy sub-domain.

Some attempts have been made to adopt the standard *SystemC* framework also in the context of electrical energy systems [9], [16], [17]. However, none of these approaches targets the modeling of EVs. The only exception is [18], that on the other hand restricts the focus to the battery management subsystem.

### B. SystemC AND ITS AMS EXTENSION

SystemC-AMS extends C/C++ with libraries to describe HW constructs [19] and analog/mixed-signal subsystems [20].

SystemC-AMS provides different abstraction levels to cover a wide variety of domains. *Timed Data-Flow* (TDF) features the modeling of discrete time processes, that are scheduled statically by considering their producer-consumer dependencies. *Linear Signal Flow* (LSF) models continuous time behaviors as mathematical relations through pre-defined primitives (e.g., integration, or delay). *Electrical Linear*

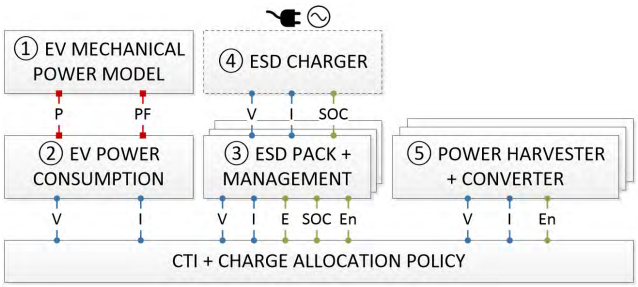



**FIGURE 1. Structure of the proposed template architecture for EV simulation in SystemC-AMS: the figure shows the typical components of the EV energy sub-domain and their interfaces, formalizing the energy and information flows in the system. Type and meaning of the signals are described in section III-B.**

*Network* (ELN) models electrical networks through the instantiation of predefined linear network primitives, e.g., resistors or capacitors. The SystemC-AMS AD solver analyzes the ELN and LSF system to derive the equations modeling system behavior, that will be solved to determine system state at any simulation time. ELN is conservative, i.e., the AD solver guarantees that energy conservation laws are satisfied.

SystemC-AMS has been applied in a number of extremely heterogeneous domains [21], [22]. However, there is no work in the literature targeting the modeling of complex and large scale mechanical systems like EVs. The only mechanical applications in the literature are MEMS systems, modeled by abstracting the dynamics equations through model order reduction or by mapping mechanical phenomena to the electrical domain [23], [24].

## III. PROPOSED FRAMEWORK

Modeling an EV requires to trace its physical and mechanical evolution, together with the energy flows and the environmental characteristics: EV power consumption (during movement) and production (during regenerative breaking) are indeed highly dependent on the mechanics of the vehicle (e.g., motor power rating and efficiency) and on its operating conditions (e.g., motor torque and angular speed). This requires to support multiple domains in the same simulation environment, i.e., mechanical and electrical energy, plus the cyber domain for the management policy.

The proposed framework builds upon the one proposed in [9], that targeted the modeling of electrical energy systems. However, that work restricts support to the DC domain, and it does not cover mechanical models. Thus, this work extends the framework to cover EVs, by formalizing the new power and information flows.

### A. FRAMEWORK FOR ELECTRICAL VEHICLE SIMULATION

Figure 1 shows the template of the architecture for the simulation of the energy subsystem of an EVs in SystemC-AMS, built by identifying the information and energy flows inside of the system that should be tracked at simulation time.

Components naturally have different roles w.r.t. the power flow, i.e., each EV includes component that either consume, generate, distribute, or store energy. This difference is reflected upon simulation: components with different roles will have different interfaces and models. This leads to a classification of components of the energy domain of an EV.

The most characteristic block is the *mechanical power model* (1), i.e., the block that reads environmental data of acceleration, speed and road slope to estimate the mechanical power demand. This block belongs to the mechanical domain, and it provides the power demand to the *electrical power model*(2), that converts the mechanical power to AC electrical power and operates as an inverter to move to the direct current (DC) domain.

Power is mostly provided by *energy storage devices* (ESDs) (3), typically batteries. Such devices are necessary to allow EV operation, and they are conveniently managed and charged from the utility grid to prolong EV lifetime (4). More than one ESD are allowed in the same system, e.g., to meet the desired voltage and current levels.

EVs may additionally feature *power sources* that harvest some quantity (e.g., weather-related phenomena) to generate additional power (5). A typical example are PV panels, that can be used to cover the exterior of the EV to prolong battery lifetime. Power sources are not mandatory, i.e., an EV may rely only on ESDs.

All the system is then connected to a charge transfer interconnect (i.e., the *CTI bus*) and managed by a charge allocation policy, that monitors the state of charge (SOC) of the ESDs and the power produced by the power sources, to determine whether the EV should be stopped and charged (6). If necessary, *converters* may be introduced to maintain compatibility of voltage levels between the CTI bus and the ESDs and the power sources.

### B. INTERFACES OF THE COMPONENTS

In this work, the focus is restricted to the energy subsystem of EVs. Thus, the most important quantities to be tracked at simulation time are the expressions of power: $P$ for AC power (in red in Figure 1), and $V$ and $I$ for DC voltage and current (in blue), respectively. In detail, the interfaces of the components are:

1) the mechanical power model reads environmental data, and shares the estimated power demand ($P$), plus a coefficient that exposes characteristics of the AC electrical curves, called power factor ($PF$, as will be detailed later);
2) the electrical power model converts such mechanical power ($P$) to DC electrical power expressed in terms of voltage ($V$) and current ($I$);
3) ESDs share their voltage ($V$), and they receive in input the current demand ($I$) of the system. Furthermore, they communicate their state of charge ($SOC$) and their nominal capacity ($E$), so that they can be monitored and activated by the charge allocation policy through an enable signal ($En$);
4) the ESD charger is connected to the utility grid (here abstracted): it takes in input the ESD voltage ($V$) and

state of charge ($SOC$), and it provides the ESD with current ($I$) to charge it whenever necessary;

5) the interface of power sources includes the supplied DC current ($I$) and voltage ($V$). Additionally, one activation signal is used to keep track of when the power source is used to feed the EV ($En$);
6) the CTI bus connects ESDs, power sources and the electrical load model. Its interface thus includes a couple of $I$ and $V$ ports per connected component, plus the ports carrying information about the status of each ESD (i.e., $SOC$, $E$ and $En$), that are used by the charge management policy to control the energy flow.

### C. FRAMEWORK IMPLEMENTATION

The framework proposed in [9] works well with the DC domain: SystemC-AMS proved to guarantee fast yet accurate simulation w.r.t. state-of-the-art tools like Matlab/Simulink, with speedups up to two orders of magnitude and a high level of accuracy.

The support of EV introduces new challenges, that must be analyzed and fixed in an effective way to avoid killing performance and accuracy:

- The main issue is the *support of the AC domain*, that relies on sinusoidal quantities and requires complex non-linear conversion circuits to and from the DC domain [25], [26]. Section III-D explains how we succeeded at modeling AC, given that SystemC-AMS does not directly support AC descriptions.
- An additional problem is introduced by the presence of *mechanical components*, since there is no work in the literature targeting the modeling of complex and large scale mechanical systems like EVs in SystemC-AMS. Section III-E presents how we achieved a good performance/accuracy trade off.

### D. EXTENDING SUPPORT TO AC COMPONENTS

In the DC domain, current and voltage have a constant direction, even if their value may change over time; thus, SystemC-AMS variables and ports can be naturally used to model their value over time.

Vice versa, in the AC domain, voltage and current have a sinusoidal nature. SystemC-AMS only supports sinusoidal curves as sources of non-conservative data or noise (i.e., the `sca_source` LSF primitive). Thus SystemC-AMS primitives can not be exploited to straightforwardly model AC quantities. Additionally, considering such a detailed representation of power would slow down simulation, while not adding de facto information w.r.t. the goal of the simulation framework, that is the overall validation of the energy flows and of the dimensioning of the energy sub-domain.

To solve this issue, we observed that a sinusoidal curve can be abstracted as its *root mean square* (RMS), i.e., a quadratic mean that can be calculated also for continuously varying functions, thus taking into account variations of the amplitude of the sinusoidal curve over time (Figure 2.a).

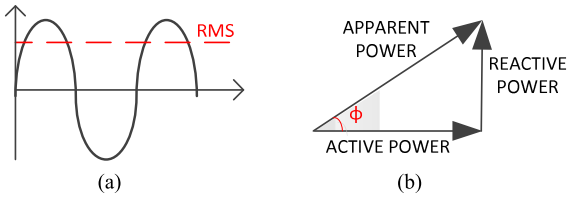



**FIGURE 2.** AC modeling in SystemC-AMS: RMS of a sinusoidal curve (a) and relationship between active and apparent power, given the presence of a phase $\phi$ between the voltage and current sinusoidal curves (b).

This implies that the involved quantities (i.e., AC power, voltage and current) can be represented by a single value and thus with standard SystemC-AMS ports and variables, as done for the DC domain.

However, this is not enough to represent AC power: AC power is determined by two sinusoidal curves, i.e., one for current and one for voltage, that may be out of phase by a degree $\phi$. This generates power that does no work at the load and that is rather wasted, but that still must be supplied by the power source, called *reactive power*. This is often the case with EVs, as they mostly behave as inductive loads, just like all mechanical motors. As a consequence, in the AC domain we must simulate the sum of active power and reactive power, called *apparent power*, that can be derived from active power by multiplying it by a factor $\cos\phi$, called also *power factor* (as evident from Figure 2.c). In case of electric motors, the power factor is typically between 0.7 and 0.9 [27].

As a result, power demand of an AC load is represented by a port $P$, representing the RMS of active power over time, and a port $PF$, exposing the corresponding power factor. Note that the latter port is necessary as the power factor is typically not constant: it may indeed vary depending on which mechanical components are active at any time. On the other hand, a model of power consumption may produce directly the apparent power, with no explicit information about the power factor. In this case, the load uses only the $P$ port, to convey the apparent power, and it thus avoids the $PF$ port, given that the evolution of power factor is not available. This allows a higher degree of flexibility in terms of models of power consumption that can be adopted.

The management of AC and of the conversion to and from DC require complex non-linear conversion circuits called *inverters* [25], [26], which exploit non-linear objects such as diodes that are not supported in the current version of SystemC-AMS. Explicitly modeling such circuitry would be useless, given the level of detail of the proposed representation of AC sinusoidal curves, and it would be a tight bottleneck for simulation speed. Interestingly, inverter datasheets measure conversion losses in terms of efficiency, i.e., of ratio between the generated AC power w.r.t. the input DC power [28], [29]. This efficiency tends to be constant and almost independent on the amount of input AC power [30], as depicted by Figure 3. Thus, it is possible to approximate AC-DC conversion as its efficiency, that is either constant or

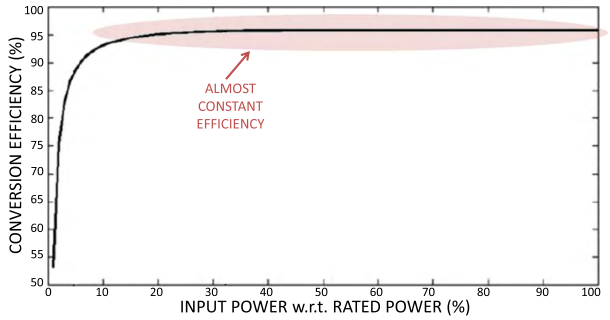



**FIGURE 3.** Typical efficiency curve of an inverter: efficiency tends to be constant and almost independent on input power, whenever this is at least 15% of rated power.

function of the sole input power and that thus can be simply modeled in SystemC-AMS.

### E. EXTENDING SUPPORT TO MECHANICAL AND PHYSICAL MODELS

SystemC-AMS does not natively cover the mechanical domain: it does not support mechanisms such as the `nature` construct of VHDL-AMS, that allows to define a physical discipline or energy domain and to describe its conservation laws [31]. SystemC-AMS is rather restricted to data flows and signals, or to electrical linear circuits. On the other hand, the goal of this work is not to accurately reproduce the mechanical behavior of an EV, but to rather have a quick estimation of power consumption given a driving cycle, so that the designer can get a quick feedback on battery lifetime and on the autonomy of the EV. Thus, we do not need a detailed mechanical model of the EV.

For this reason, we adopted a model that represents the behavior of the EV as equations [3], [32]. This model derives EV power consumption as a function of road slope, vehicle speed and acceleration over time, that can be easily modeled as C++ functions. The result is a good trade off in terms of accuracy and simulation speed: the empirical polynomial equations ensure good simulation performance, and the model takes into account mechanical phenomena, such as loss on the motor and drivetrain, that heavily impact on vehicle power consumption.

Instantaneous power consumption of an EV is dominated by propulsion power. Thus, it is possible to abstract the power consumption of the EV as a function of road slope $\theta$, EV acceleration $a$, EV mass $m$ and EV velocity $v$, such that:

$$P_{dyna} \approx (\alpha + \beta sin\theta + \gamma a + \delta v^2)mv \qquad (1)$$

where coefficients $\alpha$, $\beta$, $\gamma$ and $\delta$ depend on the vehicle rolling resistance, gradient resistance, inertia resistance, and aerodynamic resistance. Coefficients can be derived as in [3] from publicly available data or from experiments.

The model additionally takes into account the efficiency of the motor and of the drivetrain. Overall EV power demand is thus correlated to dynamic power by a factor $\eta$:

$$P_{EV} = \frac{P_{dyna}}{\eta} \text{ and } \eta = \frac{P_{dyna}}{P_{dyna} + C_0 + C_1 v + C_2 v^2 + C_3 T^2}$$

where $T$ is the motor torque and $C_0$, $C_1$, $C_2$, and $C_3$ are coefficients for constant loss, iron and friction losses, drivetrain loss, and copper loss, respectively [3].

Finally, the model supports also regenerative breaking, i.e., kinetic energy of the EV can be converted to electric energy during breaking periods:

$$P_{regen} = \epsilon T v + \zeta$$

where $\epsilon$ is regenerative force and $\zeta$ is the minimum power to generate regenerative power [3], [32].

### F. IMPLEMENTATION IN SYSTEMC-AMS

The resulting system can be easily implemented in SystemC-AMS. Each component is mapped onto one SystemC-AMS module. Interfaces are implemented as TDF ports, since the execution semantics of TDF accelerates simulation by defining a static schedule, and thus enforces an efficient interaction between components.

Even if we enforce a TDF interface, the flexibility and modularity of SystemC-AMS allows to adopt the most suitable abstraction level for the implementation of the model of each component.

When the component has a *functional model* (e.g., based on equations, as done for the mechanical power model, or the cyber control implemented inside of the CTI bus), the SystemC-AMS module is implemented as a TDF module (`SCA_TDF_MODULE`). This is the case of Figure 4.a, that shows the implementation of the EV power consumption model. Component evolution is handled by the `processing()` function, that encapsulates the C++ implementation of the model (i.e., all the equations). This function is repeatedly executed over time to evaluate the evolution of power consumption on the updated values of speed, acceleration and road slope (lines 16–22). The `initialize()` function is executed at the beginning of simulation, and it initializes the value of coefficients (lines 9–15).

*Circuit-equivalent models* are implemented as standard SystemC modules (`SC_MODULE`) encapsulating the instantiation of ELN primitives, used to map the circuit elements. Native ELN-to-TDF converters are used to convert signals between the ELN primitives and the TDF interface. This is usually the case of battery or power source models. Figure 4.b shows an example of circuit-equivalent battery model (right) and a snapshot of its SystemC-AMS implementation built by connecting ELN primitives, like capacitors (`sca_c`, line 5), current sources (`sca_isource`, line 6), voltage sources (`sca_vsource`, line 7), etc.

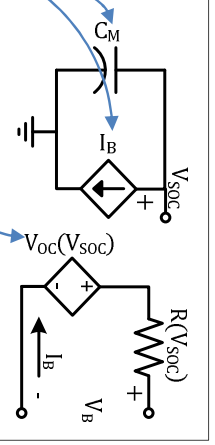
## IV. GENERATION OF DRIVING CYCLE

To generate speed, acceleration and road slope over time for a desired driving cycle, we used a strategy based on Google

```
1. SCA_TDF_MODULE(EV_mechanical){
2.   sca_tdf::sca_out<double> P;
3.   sca_tdf::sca_in<double> velocity, accel, slope;
4.   double T, vel, alpha, beta, C0, C1, ...;
5.   EV_mechanical( sc_core::sc_module_name name_){};
6.   void initialize();
7.   void processing();
8. };
9. void EV_mechanical::initialize(){
10.  alpha = 0.3210;
11.  beta = 10.1138;
12.  C0 = 5.28;
13.  C1 = 7.3927;
14.  ...
15. }
16. void EV_mechanical::processing(){
17.  vel = velocity.read();
18.  T = (alpha + beta * sin(slope.read()) + gamma * accel.read()
       + delta * exp(vel,2)) * EV_mass * vel;
19.  if(T>0)
20.    P.write(T/(C0 + C1 * vel+ C2 * exp(vel, 2) + C3 * exp(T, 2)));
21.  else  P.write(epsilon * T * vel+ zeta);
22. }
```

(a)

```
1. SC_MODULE (esd_battery){
2.   sca_tdf::sca_in<double> I;
3.   sca_tdf::sca_out<double> V, SOC, E;
4.   sca_tdf::sca_in<bool> En;
5.   sca_eln::sca_c* Cm;
6.   sca_eln::sca_tdf::sca_isource* Ib;
7.   sca_eln::sca_tdf::sca_vsource* Voc;
8.   sca_eln::sca_node n1;
9.   sca_eln::sca_node_ref gnd;
10.  ...
11.  SC_CTOR (esd_battery){
12.    Ib = new sca_isource("Ib");
13.    Ib->inp(in);
14.    Ib->p(n1);
15.    Ib->n(gnd);
16.    ...
17.  }};
18. };
```




(b)

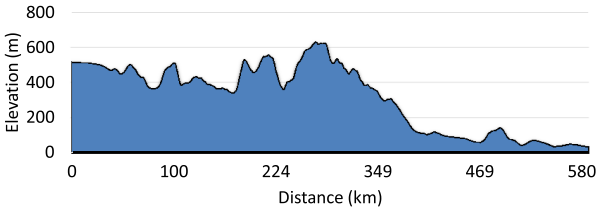
**FIGURE 4.** Snapshots of SystemC-AMS code: (a) TDF code implementing an example of mechanical model of EV power consumption and (b) ELN circuit-equivalent model of a battery.

Maps [33]. Given the departure and arrival points and the time to departure, Goole Maps finds the optimal route considering speed limit and traffic information at that time. Figure 5(a) shows an example route: the route is divided into several segments according to vehicle speed determined by traffic and speed limit of the road (red for heavy traffic, orange for medium traffic, and blue for no traffic, i.e., vehicles can accelerate until speed limit). We can estimate the average speed of each segment based on the distance of the segment and on the average driving time to pass the segment, which is reported by Google Maps. Acceleration is then derived as a byproduct of speed.

Road slope is obtained by extracting GPS data along the route, that allow to obtain the corresponding elevation data [34]. As an example, Figure 5(b) shows the elevation data of the route in Figure 5(a).

The environmental traces are then saved to files, that are loaded all along SystemC-AMS simulation.


(a)




(b)

**FIGURE 5.** Example of driving route: (a) road traffic information and (b) road slope information.

## V. SIMULATION RESULTS

This Section proves the effectiveness of the framework on two EVs: a Tesla Model 3 car, used to compare the framework w.r.t. ADVISOR (Section V-A), and a custom EV, used for design space exploration (Section V-B).

### A. CASE STUDY 1: COMPARISON WITH STATE-OF-THE-ART TOOLS

Tesla Model 3 (Figure 6) is a long-range EV produced for the mass-market equipped with:

- an internal permanent magnetic electric motor, with maximum power 202kW from 4,700 to 9,000 RPM, and maximum torque 416 Nm from 100 to 4,500 RPM respectively. The motor supports regenerative breaking, i.e., power production during the breaking phase;
- a battery pack including 4,416 4.17V 5.065Ah cells with a 46p×96s arrangement, for an overall nominal operating conditions of 233Ah and 400V. This ensures a long driving range (220km to 670km).

The resulting energy subsystem is depicted on the bottom of Figure 6. The reference driving profile is a 586km route from

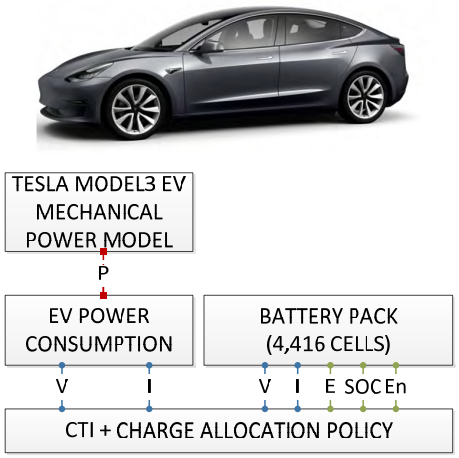



**FIGURE 6.** Tesla model 3: picture (top) and energy subsystem modeled with the proposed framework (bottom).

**TABLE 1.** Coefficients for tesla model 3.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 0.098 | $\beta$ | 10.223 | $\gamma$ | 0.989 | $\delta$ | $0.174e^{-3}$ |
| $C_0$ | 1,860 | $C_1$ | 164.068 | $C_2$ | 0.031 | $C_3$ | $0.513e^{-3}$ |
| $\epsilon$ | 0.691 | $\zeta$ | 7,790.600 | | | | |

Munich to Berlin (Figure 5). The average speed is 91 km/h, and road slope is from -2.13% to 2.23%.

#### 1) CONSTRUCTION OF THE SIMULATABLE MODELS IN ADVISOR AND SYSTEMC-AMS

We implemented the Tesla Model 3 EV both in the proposed framework and in ADVISOR, with the goal of evaluating performance and accuracy. To feed the ADVISOR model, we extracted information related to vehicle specification and driving results [35]–[37], and we adopted a battery model compatible w.r.t. the specifications in terms of capacity and operating conditions.

To reproduce the EV in the proposed framework, we characterized the power consumption model in Section III-E from the available information and from driving profiles as presented in [3]. The derived values of the coefficients are reported in Table 1. The power consumption model calculates apparent power, thus the power factor is not modeled and the *PF* port is removed.

To model the battery, we adopted a circuit equivalent model of a single cell, as presented in [38]. The single battery cell has been modeled by adopting a circuit equivalent model, implemented in SystemC-AMS ELN and characterized with information extracted solely from the datasheet of battery. The adopted model provides accurate simulation results, since it is sensitive to load frequency and current rate [38]. We additionally implemented an ideal scaling of the battery cell model according to the serial and parallel interconnection, to allow faster simulation runs and a higher flexibility in the modeling of large battery pack, so that not all cells of a large pack have to be simulated individually. The battery module also includes a battery management system, in charge of controlling the charge and discharge phases, and of avoiding over-charging and over-discharging to reduce the capacity aging degradation of battery [39].

#### 2) COMPARISON OF SIMULATION RESULTS W.R.T. ADVISOR

Figure 7 compares the simulation results of SystemC-AMS and ADVISOR. The simulation stops when the battery management system detects a battery SOC of 20%, which is a widely accepted safety bottom line to reduce battery aging. The figure shows input speed, road slope and acceleration profiles (top), computed mechanical power (middle), and battery SOC over time (bottom). The positive peaks in the output mechanical power occur when the EV enters into the highway, while the negative peak occurs when the EV brakes due to a traffic jam. This allows regenerative breaking, thus producing electrical energy and charging the battery (more evident in the zoom in in Figure 8).

Figure 7 highlights that SystemC-AMS follows well ADVISOR, as proved by the error traces reported for the computed mechanical power and the battery SOC: the average absolute error on power is 2.26%, and the error of SOC is 0.13%. The zoom in on the right hand side allows to better elaborate on the evolution of error over time. The computed mechanical power has a maximum relative error of 20%: this is caused by the mechanical model adopted in our framework, that abstracts the actual behavior of the EV (e.g., wheel inertia, impact of the transmission system and of the gear box). However, the zoom in allows to appreciate that the peak error is limited in time, and it is mostly caused by sudden changes of vehicle speed, acceleration and road slope, whose effect is not modeled as accurately as by the ADVISOR mechanical model. Thus, this level of accuracy is acceptable, given the level of detail of the adopted power model. Nonetheless, the absolute error is still limited, and the estimation of mechanical power quickly converges on values that adhere to the ADVISOR traces. In fact, the proposed framework is independent from the power model adopted: thus, future work may include an exploration of more accurate power models, to improve the accuracy of mechanical power estimation.

The other peak of relative error is at the beginning of the simulation: by looking at the absolute error, it is easy to notice that the high relative error (in the order of 12%) is caused by the low value of power consumption (around 5kW): even small variations in the estimation of power (i.e., less that 0.5kW) are indeed relatively significant while they are limited if compared to the EV peak power consumption (48kW).

It is interesting to note that, in correspondence to the peak errors on mechanical power, the error on battery SOC is still limited and almost flat (around 0.15%). Indeed, the longer temporal dynamics of the battery partially compensates the error on power demand, thus achieving a higher accuracy on the estimation of the battery SOC. Thus, the energy subsystem is simulated with a level of accuracy that satisfies the requirements for our framework.

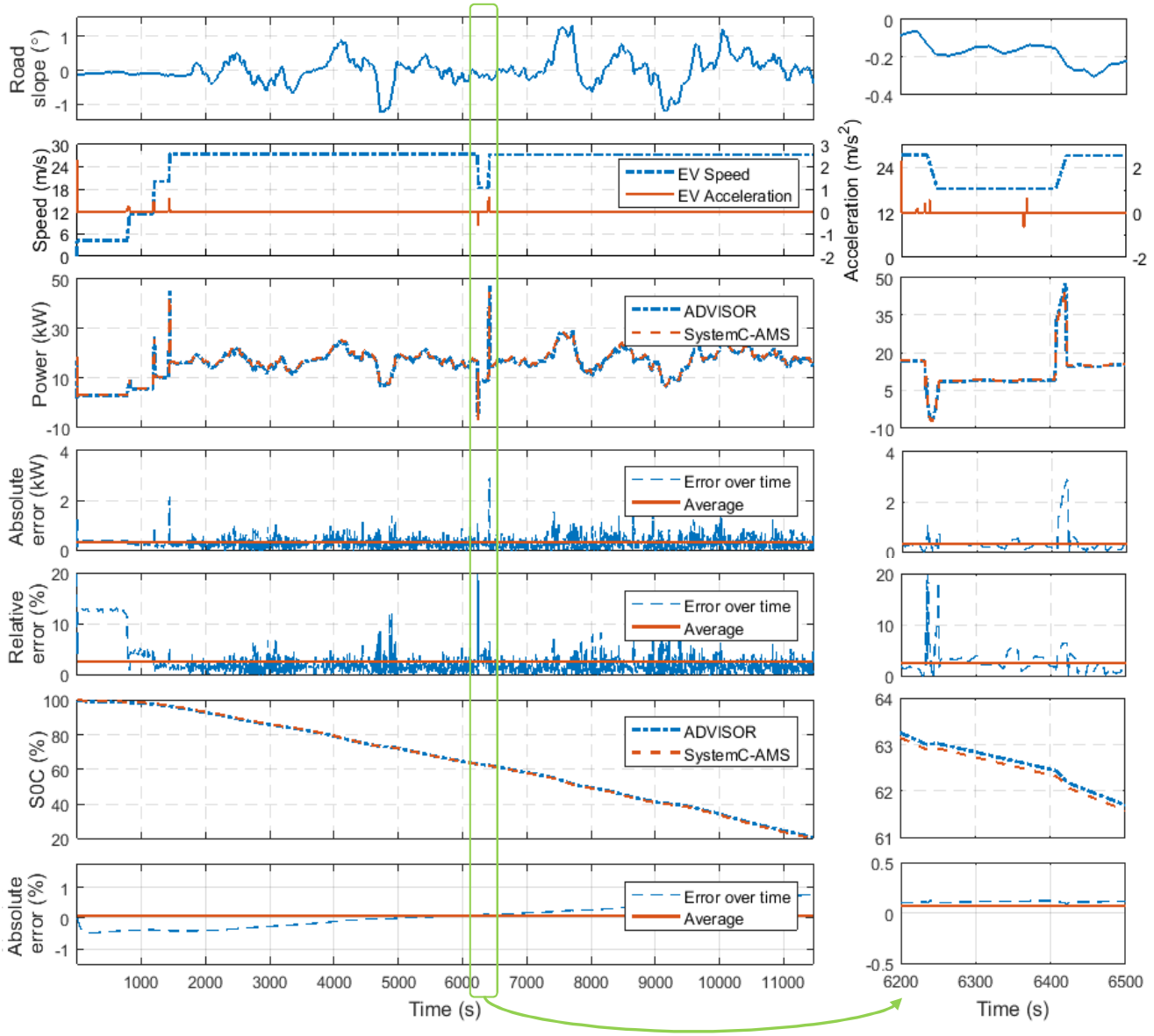



**FIGURE 7.** Simulation results of SystemC-AMS and ADVISOR: speed, road slope and acceleration profiles (top), computed mechanical power (middle), and battery SOC over time (bottom) with the corresponding error. The zoom in allows to better appreciate the evolution of error as a result of variations of the iput environmental values.

On the other hand, SystemC-AMS simulation is 438 times faster than ADVISOR: runtimes are 0.15 seconds and 71.04 seconds, respectively. This speedup is once again strictly correlated with the complexity of the mechanical model adopted in this work, and by the structure of the simulation framework. SystemC-AMS is organized in 7 SystemC-AMS modules (4 of which devoted to battery modeling, to have an accurate simulation, and only 1 for the mechanical model) and includes only 12 ELN primitives (i.e., 7 ELN blocks and 5 ELN nodes). The efficient activation of TDF modules, based on a static scheduling, and the limited number of ELN primitives guarantees fast simulation and a low overhead of both the solver and the scheduler. Vice versa, the ADVISOR system is implemented as over 3,000 Simulink blocks, including 114 switches, 9 S-Functions (invoking external Matlab and C code) and 28 lookup tables, that heavily impact on simulation performance. Interestingly, both SystemC-AMS and ADVISOR use the same solver configuration, i.e., a first order solver based on Euler's method with a fixed time step of 1s.

These results show that the proposed SystemC-AMS framework exhibits both a high level of accuracy and a speedup of more than three orders of magnitude, thus enhancing EV design and allowing quick design space exploration in early design stages.

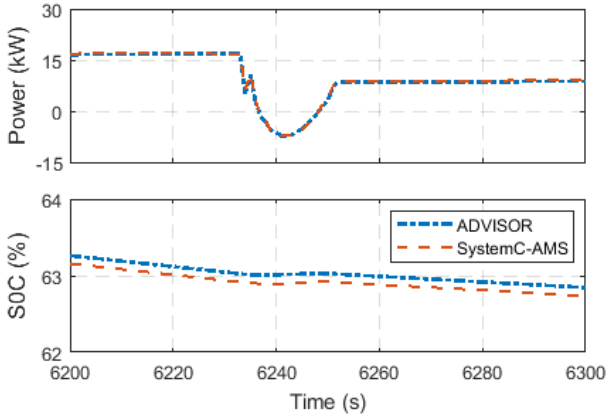



**FIGURE 8.** Zoom in of the evolution of computed mechanical power (top) and battery SOC over time (bottom) to highlight the impact of regenerative breaking on battery SOC.

**TABLE 2.** Coefficients for the custom EV.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 0.321 | $\beta$ | 10.114 | $\gamma$ | 1.075 | $\delta$ | 0.000 |
| $C_0$ | 5.280 | $C_1$ | 7.393 | $C_2$ | 20.620 | $C_3$ | 0.002 |
| $\epsilon$ | 460.500 | $\zeta$ | 355.919 | | | | |

### B. CASE STUDY 2: SUPPORT FOR DESIGN SPACE EXPLORATION

The second case study is a smaller custom EV, i.e., a small dune buggy powered by an electric motor. The EV can host up to 10 PV modules: they are indeed an ideal power source for EVs, since they are light-weight and durable, and they allow to prolong EV autonomy. Given the small area covered by the PV modules (i.e., rooftop and back panel), they are not enough to fully operate the EV, but they can be exploited to charge the battery when the EV is parked and not plugged to any charging facility.

The EV characteristics are available in [3]:

- the EV is powered by a 8-pole brushless motor, with continuous power rate 1.5kW with a 7.17Nm torque. The motor supports regenerative breaking, with peak current 70A;
- the EV is equipped with a battery pack composed by NCR18650B 3.7V 3400mAh rechargeable Li-ion cylindrical cells;
- PV modules generate 20W each at standard conditions (i.e., 1000W/m$^2$ irradiance and 25°C temperature) [40]. The PV modules are connected to a solar charge controller, that applies a Maximum Power Point MPP) tracking algorithm, i.e., that determines the operating voltage over time by trying to maximize output power.

#### 1) SystemC-AMS IMPLEMENTATION

The resulting energy subsystem is depicted on the bottom of Figure 9. The EV power model has been derived as explained in Section III-E, and the resulting coefficients are shown in Table 2. The power consumption model calculates apparent power, thus the power factor is not modeled and the *PF* port is removed.

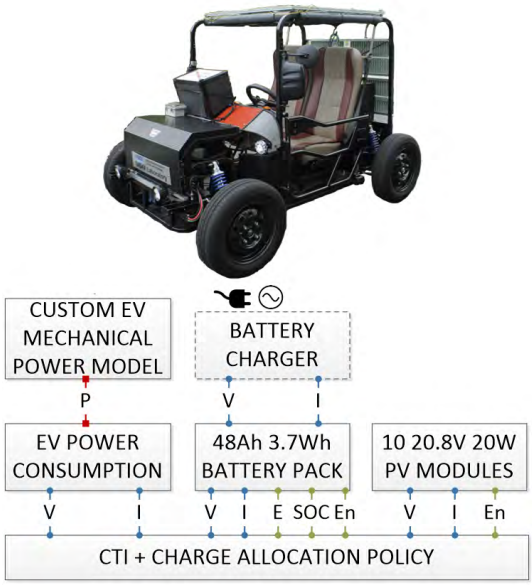



**FIGURE 9.** The custom EV: picture (top) and energy subsystem modeled with the proposed framework (bottom).

The battery pack has been modeled with a circuit-equivalent battery cell model and scaled-up ideally based on the battery pack inner configuration, as described for the former case study. The battery charger adopts a Constant Current-Constant Voltage (CC-CV) protocol, that has been optimized to ensure an optimal tradeoff between aging degradation of capacity and quality of service (e.g., smallest average SOC and smallest possible charging current during constant current phase) [41]. The maximum SOC is set to 90%, while the battery is considered discharged when the SOC is 20%.

The PV modules are modeled with a functional model, that directly extracts the MPP and that is built directly from the sole datasheet information [42]. Given in input the current vs. voltage graph available on the datasheet [40], the model identifies the corresponding maximum power points. Such points are then used as a mapping of irradiance values to output power, and fitted to a polynomial curve that can be easily implemented in SystemC-AMS. It is important to note that more accurate models might have been adopted, like the circuit-equivalent model in [43], that can be easily modeled in with ELN primitives.

Finally, the AC to DC conversion has been implemented with fixed efficiency, by observing that, whenever input power is at least 15% of the rated power, efficiency can be approximated as constant [30].

#### 2) DESIGN SPACE EXPLORATION

We adopted the proposed framework to evaluate alternative configurations in terms of number of PV modules and of battery cells. The constraint is that all solutions must have the same initial capital cost to buy the components (i.e., $750,00). In this way we created 6 different configurations. Configuration 1 is the reference configuration, depicted in Figure 9.

**TABLE 3. Configurations considered during the design space exploration analysis. Configuration 1 is considered the reference one.**

| | PV modules (#) | Battery cells (#) | Nominal capacity (Ah) | Nominal voltage (V) |
|---|---|---|---|---|
| 1 | 10 | 125 (5 × 25) | 17.0 | 92.5 |
| 2 | 8 | 135 (5 × 27) | 17.0 | 99.9 |
| 3 | 6 | 150 (6 × 25) | 20.4 | 92.5 |
| 4 | 4 | 162 (6 × 27) | 20.4 | 99.9 |
| 5 | 2 | 175 (7 × 25) | 23.8 | 92.5 |
| 6 | 0 | 189 (7 × 27) | 23.8 | 99.9 |

**TABLE 4. Capital cost, weight and lifetime of PV panel, and battery cells in custom EV.**

| | Cost ($) | Weight (g) | Lifetime (Years) |
|---|---|---|---|
| PV Panel Module | 24 | 2200 | 20 |
| Battery Cell | 4 | 55 | 3 |

All other configurations are obtained from configuration 1 by decreasing the number of PVs and by compensating with additional battery cells. Table 3 reports the main characteristics of each configuration, in terms of number of PV modules and battery cells (with their topology, in terms of parallel×series), nominal capacity and nominal voltage. Nominal battery voltage is almost constant across all configurations to match the operating voltage of the EV motor: for this reason, batteries are organized in series strings of 25 to 27 cells, so to sum up voltages, and strings are then connected in parallel.

Table 4 indicates the useful information about the PV modules and the battery cells, including unit cost, weight and expected lifetime.

We simulated all configurations on a typical 24-hours driving cycle emulating a daily commute in Munich: going to work, charging the battery through the PV panels during working hours, going back home, and charging the EV connected to the grid over night.

Figure 10 shows a snapshot of the evolution of the EV for configuration 1 during two hours in the morning, from when the vehicle leaves the house. While driving, the EV consumes power proportionally to the characteristics of the road (i.e., speed, acceleration, and road slope, top of the Figure). The battery SOC (middle) and voltage (bottom) decrease correspondingly, since the EV is powered by the battery. Once that that the car is parked, the battery is charged by the PVs: power consumption is zero, and the battery SOC and voltage slowly increase.

Figure 11 compares the 6 configurations in terms of their composition and their performance. The figure reports: the number of PV modules and battery cells, the weight of the energy subsystem (dominated by the PV modules and by the battery cells), the residual SOC before and after going home, and the daily cost of each configuration. All values are normalized w.r.t. configuration 1.

The most evident result is that configuration 6, i.e., the one with no PV, does not complete the driving cycle, since the battery SOC ends before reaching home. Indeed, the battery

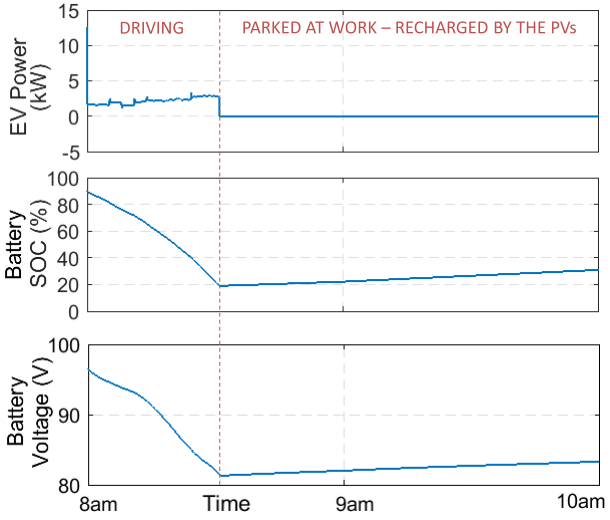



**FIGURE 10. Snapshot of the evaluation of EV power consumption (top), battery SOC (middle), and battery voltage (bottom) for configuration 1.**

SOC before leaving work is the lowest of all (39.5%), and it is too low to complete the path to home. Thus, even if PVs have a small rated power, their contribution can not be ignored as they can charge the battery while the EV is parked and they allow to complete the driving cycle. For this reason, most metrics are not available for configuration 6.

For all other configurations, residual SOC at home, before plugging to the grid, is mostly influenced by the number of battery cells: the higher the capacity, the longer the lifetime. The weight of the energy subsystem has a lower impact: the heaviest configuration (i.e., configuration 6 with 28.875 Kg) impacts for only the 4% of the total EV weight (the sole mechanical parts weigh around 600Kg).

As an additional evaluation metric we adopt *total daily cost*, including:

- electricity cost for charging the battery pack, estimated as $c_{el} = \int e(t)P(t)dt$ where $e(t)$ is the instantaneous electricity cost over time (modeled as a stepwise function, with different costs at different times of the day) and $P(t)$ is the instantaneous charge power;
- wear cost, due to aging, of each component of the energy subsystem, estimated as

$$c_{wear} = \frac{\text{initial capital cost}}{\text{lifetime} \times 365}$$

- operating and maintenance (O&M) cost for PV modules, estimated as 65$/kW per year. We added such O&M cost to the wear cost of PV panel.

PV wear cost and battery wear cost linearly depend on the number of units in each configuration. Electricity cost mainly depends on two main factors, i.e., the nominal total capacity of battery pack and the SOC of battery pack when the EV is plugged to the charger at the end of the day. This cost is almost constant for configurations 1 to 4. Surprisingly, configuration 5 has a sensibly higher electricity cost. This configuration features only two PVs, that do not charge enough the EV

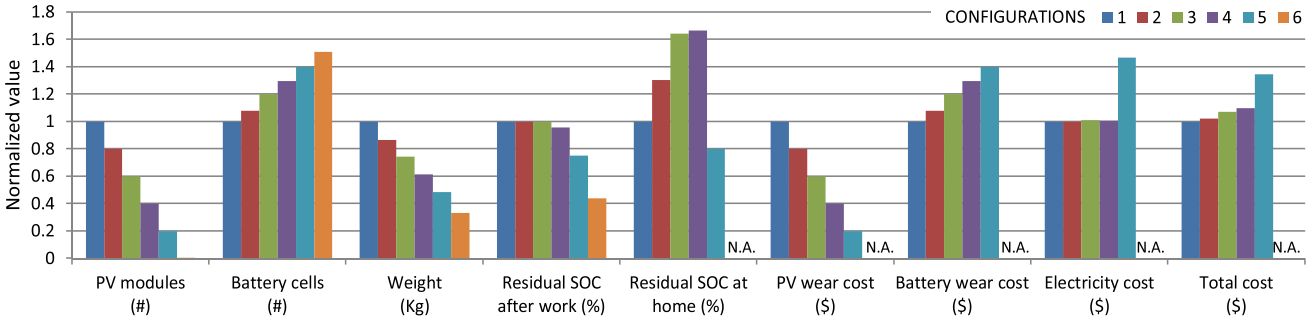



**FIGURE 11.** **Results of the design space exploration: the six configurations are compared in terms of number of PV modules and of battery cells, weight, residual SOC in two different moments of the day, and costs. Values are normalized w.r.t. configuration 1. Not all metrics are available for configuration 6, since this configuration does not complete the driving cycle.**

during the working hours. Thus, the battery SOC when the EV leaves to go home is sensibly lower (67%) than the other configurations, that reach almost full charge (89% in average, considered that the charging policy stops at 90% for aging reasons). This implies that configuration 5 requires a longer charging phase, and this impacts on the total electricity cost: for these reasons, configuration 5 is not an optimal one for this EV.

As a result, the designer may consider three configurations as optimal outputs of the design space exploration: configuration 1, that minimizes total daily cost, and configurations 3 and 4, that achieve a higher residual SOC (+65.56% in average) and impact on battery aging with a minimal impact on cost (+8.61%) w.r.t. configuration 1. It is important to note that simulations lasted in average 0.91s, thus the design space exploration allowed to determine optimal configurations and their pros and cons in a few seconds overall.

## VI. CONCLUSIONS AND PERSPECTIVE

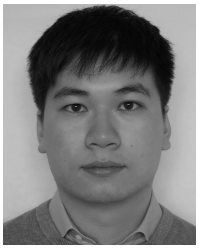
This work proved to successfully extend a SystemC-AMS-based framework for the modeling of electrical energy systems to EVs. The experimental results proved a good level of accuracy both in the estimation of power consumption (error about 2% w.r.t. the state-of-the-art tool ADVISOR) and of battery dynamics (error lower than 1%). This ensures an effective estimation of component behavior. The achieved speed-up w.r.t. ADVISOR (of more than two orders of magnitude) enhances the design process, and allows to explore multiple alternatives to determine the most effective configuration of the energy components in terms of autonomy, component aging and cost, as proved by the second case study. Thus, the adoption of the proposed framework allows to effectively improve the dimensioning of the energy subsystem and the driving range of the EV under design.

Future work should focus on identifying a more accurate model of physical aspects, to reduce approximation margins while still allowing good simulation performance. Possible solutions are either refining the model for mechanical power, e.g., by increasing the order of polynomials or by considering more environmental inputs like wind, or reducing the mechanical aspects to domains supported by the SystemC-AMS standard, e.g., by building electrical equivalent models, that emulate the behavior of non-electrical phenomena through electrical linear components (like resistors or capacitors).

## REFERENCES

[1] M. Lukasiewycz *et al.*, "Cyber-physical systems design for electric vehicles," in *Proc. Euromicro DSD*, 2012, pp. 477–484.
[2] K. Tehrani and O. Maurice, "A cyber physical energy system design (CPESD) for electric vehicle applications," in *Proc. IEEE SoSE*, Jun. 2017, pp. 1–6.
[3] N. Chang, D. Baek, and J. Hong, "Power consumption characterization, modeling and estimation of electric vehicles," in *Proc. IEEE/ACM ICCAD*, Nov. 2014, pp. 175–182.
[4] K. L. Butler, M. Ehsani, and P. Kamath, "A MATLAB-based modeling and simulation package for electric and hybrid electric vehicle design," *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1770–1778, Nov. 1999.
[5] L. Gauchia and J. Sanz, "Study of the effect of adverse external conditions on electric vehicles control: Adaptive energy management as a solution," *IFAC Proc. Volumes*, vol. 45, no. 21, pp. 460–465, 2012.
[6] The Mathworks. (2005). *Electric Vehicle Configured for HIL*. [Online]. Available: https://www.mathworks.com/help/physmod/elec/examples/electric-vehicle-configured-for-hil.html
[7] S. Alegre, J. V. Míguez, and J. Carpio, "Modelling of electric and parallel-hybrid electric vehicle using MATLAB/simulink environment and planning of charging stations through a geographic information system and genetic algorithms," *Renew. Sustain. Energy Rev.*, vol. 74, pp. 1020–1027, Jul. 2017.
[8] Y. Hirano, S. Inoue, and J. Ota, "Model based performance development of a future small electric vehicle using modelica," in *Proc. IEEE SICE*, Jul. 2015, pp. 1369–1374.
[9] S. Vinco, A. Sassone, F. Fummi, E. Macii, and M. Poncino, "An open-source framework for formal specification and simulation of electrical energy systems," in *Proc. IEEE/ACM ISLPED*, Aug. 2014, pp. 287–290.
[10] K. B. Wipke, M. R. Cuddy, and S. D. Burch, "ADVISOR 2.1: A user-friendly advanced powertrain simulation using a combined backward/forward approach," *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1751–1761, Nov. 1999.
[11] T. Markel *et al.*, "ADVISOR: A systems analysis tool for advanced vehicle modeling," *J. Power Sources*, vol. 110, no. 2, pp. 255–266, 2002.
[12] A. Rousseau, P. Sharer, and F. Besnier, "Feasibility of reusable vehicle modeling: Application to hybrid vehicles," SAE Tech. Paper 2004-01-1618, 2004.
[13] G. Cole. Simple Electric Vehicle simulation (SIMPLEV) Version3.1. DOE Idaho National Engineering Laboratories. Accessed: Feb. 20, 2019. [Online]. Available: www.inl.gov/article/electric-vehicle-research/
[14] M. Ehsani, Y. Gao, and K. L. Butler, "Application of electrically peaking hybrid (ELPH) propulsion system to a full-size passenger car with simulated design verification," *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1779–1787, Nov. 1999.
[15] Y. Hirano, S. Inoue, and J. Ota, "Model-based development of future small EVs using modelica," in *Proc. Int. Modelica Conf.*, 2014, pp. 63–70.
[16] J. M. Molina, X. Pan, C. Grimm, and M. Damm, "A framework for model-based design of embedded systems for energy management," in *Proc. IEEE MSCPES*, May 2013, pp. 1–6.

[17] J. M. Molina, X. Pan, C. Grimm, and M. Damm, "A framework for model-based design of embedded systems for energy management," in *Proc. IEEE MSCPES*, May 2013, pp. 1–6.
[18] C. Scavongelli, F. Francesco, S. Orcioni, and M. Conti, "Battery management system simulation using SystemC," in *Proc. IEEE WISES*, Oct. 2015, pp. 151–156.
[19] *IEEE Standard for Standard SystemC Language Reference Manual*, IEEE Standard 1666-2011, 2012, pp. 1–638.
[20] *IEEE Standard for Standard SystemC(R) Analog/Mixed-Signal Extensions Language Reference Manual*, IEEE Standard 1666.1-2016, 2016, pp. 1–236.
[21] V. Fernandez, E. Wilpert, H. Isidoro, C. Ben Aoun, and F. Pêcheux, "SystemC-MDVP modelling of pressure driven microfluidic systems," in *Proc. IEEE MECO*, Jun. 2014, pp. 10–13.
[22] L. Gil and M. Radetzki, "SystemC AMS power electronic modeling with ideal instantaneous switches," in *Proc. IEEE/ECSI FDL*, vol. 978, Oct. 2014, pp. 1–8.
[23] T. Machne *et al.*, "UVM-SystemC-AMS based framework for the correct by construction design of MEMS in their real heterogeneous application context," in *Proc. IEEE ICECS*, Dec. 2014, pp. 862–865.
[24] F. Cenni, O. Guillaume, M. Diaz-Nava, and T. Maehne, "SystemC-AMS/MDVP-based modeling for the virtual prototyping of MEMS applications," in *Proc. IEEE DTIP*, Apr. 2015, pp. 1–6.
[25] R. Majumder, A. Ghosh, G. Ledwich, and F. Zare, "Power management and power flow control with back-to-back converters in a utility connected microgrid," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 821–834, May 2010.
[26] T. R. Kuphaldt, *Lessons in Electric Circuits, Volume II–AC*, vol. 2. Washington, DC, USA: ibiblio, 2007.
[27] The Engineering Toolbox. *Power Factor–Inductive Load*. Accessed: Dec. 11, 2018. [Online]. Available: www.engineeringtoolbox.com/power-factor-electrical-motor-d_654.html
[28] SolarEdge. (2017). *SE4K–SE10K Datasheet*. Accessed: Dec. 11, 2018. [Online]. Available: www.solaredge.com/sites/default/files/se-three-phase-inverter-datasheet.
[29] Meanwell. (2016). *A-301/302-150 Series Datasheet*. Accessed: Dec. 11, 2018. [Online]. Available: www.meanwell-bg.com/files/M2017H1/A300-150-SPEC.PDF
[30] T. Khatib, "Optimization of a grid-connected renewable energy system for a case study in Nablus, Palestine," *Int. J. Low-Carbon Technol.*, vol. 9, no. 4, pp. 311–318, 2013.
[31] M. Schubert. (2017). *VHDL-AMS Fundamentals*. Accessed: Feb. 20, 2019. [Online]. Available: https://hps.hs-regensburg.de
[32] J. Hong, S. Park, and N. Chang, "Accurate remaining range estimation for electric vehicles," in *Proc. ACM ASP-DAC*, 2016, pp. 781–786.
[33] *Google Maps*. Accessed: Feb. 24, 2019. [Online]. Available: https://www.google.com/maps
[34] *Geocontext–Center for Geographic Analysis*. Accessed: Feb. 24, 2019. [Online]. Available: http://www.geocontext.org
[35] *Model 3—Vehicle Specifications, Tesla Press Information*. Accessed: Feb. 20, 2019. [Online]. Available: https://www.tesla.com/presskit#model3
[36] Tesla Model 3 75 261 ps. (2017). *Technical Specifications and Performance Figures, Zeperfs*. Accessed: Feb. 20, 2019. [Online]. Available: http://www.zeperfs.com/en/fiche7083-tesla-model-3-75.htm
[37] *Tesla Model 3 & Chevy Bolt Battery Packs Examined, CleanTechnica*. Accessed: Dec. 13, 2018. [Online]. Available: https://cleantechnica.com/2018/07/08/tesla-model-3-chevy-bolt-battery-packs-examined/
[38] Y. Chen, E. Macii, and M. Poncino, "A circuit-equivalent battery model accounting for the dependency on load frequency," in *Proc. IEEE DATE*, Mar. 2017, pp. 1177–1182.
[39] M. A. Hannan, M. M. Hoque, A. Hussain, Y. Yusof, and P. J. Ker, "State-of-the-art and energy management system of Lithium-Ion batteries in electric vehicle applications: Issues and recommendations," *IEEE Access*, vol. 6, pp. 19362–19378, 2018.
[40] Solar Electric Supply Inc. (2018). *420J 20W Photovoltaic Module*. [Online]. Available: www.solarelectricsupply.com/solar-electric-supply-ses-420j-solar-panels-540
[41] Y. Chen, S. Vinco, E. Macii, and M. Poncino, "Fast thermal simulation using SystemC-AMS," in *Proc. ACM GLSVLSI*, 2016, pp. 427–432.
[42] S. Vinco, Y. Chen, E. Macii, and M. Poncino, "A unified model of power sources for the simulation of electrical energy systems," in *Proc. ACM GLSVLSI*, 2016, pp. 281–286.
[43] A. Bauer, J. Hanisch, and E. Ahlswede, "An effective single solar cell equivalent circuit model for two or more solar cells connected in series," *IEEE J. Photovolt.*, vol. 4, no. 1, pp. 340–347, Jan. 2014.



**YUKAI CHEN** received the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 2018, where he is currently a Postdoctoral Research Fellow. His current research interest includes computer-aided design for integrated circuits and electrical energy systems, with particular emphasis on the modeling and simulation of extra-functional properties.

**DONKYU BAEK** received the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, in 2017. He is currently a Postdoctoral Research Fellow with the Politecnico di Torino, Italy. His current research interests include low-power electric vehicles and delivery drones.

**JAEMIN KIM** received the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Seoul National University, in 2018. He is currently an Assistant Professor with the Department of Electronics Engineering, Myongji University. His current research interests include low-power embedded systems and electric vehicles.

**SANTA DI CATALDO** received the Ph.D. degree in systems and computer engineering from the Politecnico di Torino, Turin, Italy, in 2011, where she has been an Assistant Professor (tenure track), since 2017. Her main research interests include image processing for biological and medical imaging applications, artificial intelligence, and heterogeneous data integration, including techniques for pattern recognition, and automated feature extraction and classification.

**NAEHYUCK CHANG** (F'12) is currently a Full Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology. His current research interests include low-power embedded systems and the design automation of things. He is an ACM Fellow and an IEEE Fellow for the contribution to low-power design.

**ENRICO MACII** is currently a Full Professor with the Politecnico di Torino, Turin, Italy. In the last few years, he has been growingly involved in projects focusing on the development of new technologies and methodologies for smart cities and bioinformatics. He has authored over 450 scientific publications in his research areas. His current research interest includes the design of electronic circuits and systems, with particular emphasis on low-power consumption, optimization, testing, and formal verification.

**SARA VINCO** received the Ph.D. degree in computer science from the University of Verona, Verona, Italy, in 2013. She has been an Assistant Professor (tenure track) with the Politecnico di Torino, Turin, Italy, since 2018. Her current research interests include energy efficient electronic design automation and techniques for the simulation and the validation of heterogeneous embedded systems. She is a member of the Editorial Board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I.

**MASSIMO PONCINO** (F'18) is currently a Full Professor with the Politecnico di Torino. He has authored or co-authored over 300 journal and conference papers. His current research interest includes several aspects of the design automation of digital systems, with particular emphasis on the modeling and optimization of low-power systems.